

AD-A265 646



DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1 AGENCY USE ONLY (Leave blank)	2 REPORT DATE April 1993	3 REPORT TYPE AND DATES COVERED professional paper
4 TITLE AND SUBTITLE CONFIGURATION OPTIMIZATION OF MOBILE MANIPULATORS WITH EQUALITY CONSTRAINTS USING EVOLUTIONARY PROGRAMMING	5 FUNDING NUMBERS In-house funding	
6 AUTHOR(S) B. Andersen, J. McDonnell, W. Page		8 PERFORMING ORGANIZATION REPORT NUMBER
7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Command, Control and Ocean Surveillance Center (NCCOSC) RDT&E Division San Diego, CA 92152-5001		
9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Command, Control and Ocean Surveillance Center (NCCOSC) RDT&E Division San Diego, CA 92152-5001		10 SPONSORING/MONITORING AGENCY REPORT NUMBER
11 SUPPLEMENTARY NOTES		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A mobile manipulator generally possesses kinematic redundancy, which requires some method for selecting the configuration appropriate to the task. Evolutionary Programming (EP) has been used successfully to select an appropriate configuration using a large penalty on the end-point position to force the end-point to the goal position for the task. This paper demonstrates that using an equality constraint for the end-point position results in much faster convergence to approximately the same final configurations. The reasons for this increase in speed are examined using a simplified model to determine the relative size of their effects.

93-12978



93 6 09 009

Published in *Proceedings of the First Annual Conference on Evolutionary Programming*, Feb 1993, pp. 71-79.

14 SUBJECT TERMS kinematic redundancy mobile manipulator evolutionary programming (EP)		15 NUMBER OF PAGES
		16 PRICE CODE
17 SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18 SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19 SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED
		20 LIMITATION OF ABSTRACT SAME AS REPORT

Configuration Optimization of Mobile Manipulators with Equality Constraints using Evolutionary Programming

Brian Andersen
John McDonnell
Ward Page

Naval Command, Control and Ocean Surveillance Center RDT&E Division

Abstract

A mobile manipulator generally possesses kinematic redundancy, which requires some method for selecting the configuration appropriate to the task. Evolutionary Programming (EP) has been used successfully to select an appropriate configuration using a large penalty on the end-point position to force the end-point to the goal position for the task. This paper demonstrates that using an equality constraint for the end-point position results in much faster convergence to approximately the same final configuration. The reasons for this increase in speed are examined using a simplified model to determine the relative size of their effects.

Introduction

The use of mobile manipulators is becoming more prevalent on the factory floor, undersea, and in space applications. Most of these systems are teleoperated or telepresence systems [1], relying upon the operator to eliminate any kinematic redundancies through direct isomorphic mappings. To make these systems more autonomous, the kinematic redundancies must be resolved without requiring operator intervention.

Various methods have been proposed to search for the best configuration of a mobile manipulator for a given task. These methods include gradient descent searches of the joint-space of the manipulator used by Pin and Culioli [2], and proposed by Burdick and Seraji [3], simu-

lated annealing used by Carriker *et al* [4], and EP, an evolutionary search strategy, used by McDonnell *et al* [5]. The EP technique was found to be an effective search technique that avoided local optima.

The cost function for optimization used in [5] imposed a high cost on deviations of the end-point position from the goal position to force the end-point to be at the required position for the task. Unfortunately, this high cost relative to the remaining elements of the cost function results in a search surface that is difficult for the EP algorithm to traverse efficiently. This paper proposes constraining the end-point position to the goal position and searching the remaining smaller-dimensional space. In addition, this paper examines the effect of the ratio of weights for different elements of the cost function on the efficiency of the EP search algorithm.

Optimization Criteria

The ultimate configuration of the manipulator is a direct function of the choice of the optimization criteria. This criteria must take into account the end-point position, obstacles in the environment, static torques required at the joints due to external forces and the weights of the links, and the ability of the manipulator to perform tasks in a particular configuration, known as the manipulability.

This section summarizes the criteria developed in [2] and [5] to determine the cost associated with mobile manipulator configurations.

The end-point position of the manipulator is the most important factor in the cost of a configuration since the manipulator will be unable to perform the task if it is not at the goal position. The cost associated with the end-point position is

$$C_{goal} = \|X_d - X_e\|$$

where

X_d = Desired Cartesian tip position

X_e = Actual Cartesian tip position

It is essential that the manipulator avoid any obstacles that may be present in its workspace. To do this, points of avoidance are specified along the links of the manipulator. The cost associated with obstacle avoidance is

$$C_{obs} = \sum_{i=0}^l \sum_{j=0}^m \frac{1}{\|X_{a,i} - X_{o,j}\|^2}$$

where

$X_{a,i}$ = Cartesian position of the i^{th} point of avoidance

$X_{o,j}$ = Cartesian Position of the j^{th} obstacle

l = Number of points of avoidance

m = Number of obstacles

The static joint torques required in a given configuration are a function of the external forces applied and the weights of each link. Any force on the manipulator can be mapped to the static joint torques required for equilibrium using the Jacobian of the manipulator at the point where the force is applied. This mapping is given by

$$\tau_m = J^T(\theta) F$$

where

τ_m = Joint torque vector

J = Jacobian of the manipulator

F = Force applied to manipulator

Therefore, the static joint torques required are given by

$$\tau_m = \sum_{i=1}^N M_i F_i + \sum_{j=1}^n J_j f_j$$

where

M_i = Jacobian associated with the center of mass of the i^{th} link

F_i = Force of the weight of the i^{th} link

J_j = Jacobian associated with the location of the j^{th} point load

f_j = j^{th} point load

N = Number of links

n = Number of point loads

The cost associated with the static torques at the joints is given by

$$C_{torq} = \frac{1}{N} \sum_{i=1}^N \left(\frac{\tau_i}{\tau_{max,i}} \right)^2$$

where

τ_i = Torque on i^{th} joint

$\tau_{max,i}$ = Maximum torque on i^{th} joint

The final element of the cost function is the manipulability, which is designed to avoid configurations near joint-space singularities where the manipulator is difficult to control. The cost associated with manipulability is given by

$$C_{man} = \frac{1}{\text{Det}[JJ^T]}$$

The complete cost function can be formulated as a weighted sum of the elements of the individual cost functions. This is given by

$$Cost = \alpha C_{goal} + \beta C_{obs} + \gamma C_{man} + \epsilon C_{torq}$$

As Pin [2] has pointed out, this cost function can be trapped in local minima when using a gradient descent algorithm. Therefore, the EP algorithm is an attractive alternative.

The EP Algorithm

The evolutionary programming algorithm performs a search by methods analogous to those found in natural evolution. The algorithm involves generating an initial population, mutating this population, determining the fitness of each element of the population, competing elements of the population against each other, and selecting the most fit elements of the population to survive to serve as the parent vectors for the next generation.

EP can be described by the following seven-step procedure, similar to the one found in Fogel [6].

1. Generate an initial population of N configuration vectors.
2. Mutate each of the N parent configuration vectors to create a population of size $2N$.
3. Determine the cost of each of the $2N$ configuration vectors.
4. Compete each member of the population against the same number of randomly-chosen opponents, assigning a win to the one with the lowest score.
5. Sort the population by decreasing number of wins.
6. Remove the N configurations with the lowest number of wins.

7. Continue with Step 2.

The Mobile Manipulator

The mobile manipulator used in these simulations is shown in Figure 1. It consists of a platform that is free to move in one dimension with a three-link planar manipulator mounted on it. The link lengths are 1.0, 1.0, and 0.5 units, respectively. Each link has a density of 10 units per unit length, and the center of gravity is at the midpoint of each link. The height of the first joint of the manipulator is 0.5 units. An external force of 0.5 units is applied acting downward at the tip of the manipulator.

The size of the mutation for each element of the population is a function of the generation number. This allows the search to cover a large range of possible configurations and still to refine the configuration vectors as the search progresses. The size of the mutations is given by

$$\delta = \mu e^{-0.001\tau k}$$

where

- δ = Size of the mutation
- μ = $N(0,1)$ random variable
- τ = $U(0,1)$ random variable
- k = Generation number

The goal position for the search was located at the Cartesian coordinates (5,1.5), as designated in Figure 1 by the filled rectangle. One obstacle was placed at the Cartesian coordinates (5,1.0), as designated in Figure 1 by the circle. One point of avoidance was chosen at the midpoint of the second link, as designated in Figure 1 by the cross. The weights in the cost function were chosen to be 100 on C_{goal} , 10 on

C_{obs} , 1 on C_{torq} , and 1 on C_{man} ($\alpha = 100$, $\beta = 10$, $\gamma = 1$, and $\epsilon = 1$).

Two different approaches were compared. The first approach used a configuration vector consisting of the cart position and the angle of each of the links relative to the horizontal, given by

$$p_1 = [x, \theta_1, \theta_2, \theta_3]$$

This representation required four elements in each configuration vector and relied on a large weight on C_{goal} to force the end-point to the goal position.

The second approach used a configuration vector that consists of only the angles of the outer two links of the manipulator, given by

$$p_2 = [\theta_2, \theta_3]$$

The cart position and inner joint angle were calculated so that the end-point was at the goal position. Therefore, instead of requiring four elements in the configuration vector, only two were required.

A complication of employing the second approach is the cart position and first angle are not uniquely determined by the outer two angles, but can take on two distinct sets of values. The values used in a given configuration were chosen randomly with equal probability to allow the search to cover as large a portion of the space as possible.

The best configuration obtained after 5000 iterations of the EP algorithm are shown in Figures 1 and 2 for the first and second approaches, respectively. The second approach results in a slightly better configuration than the first, although the configurations are approximately the same.

However, a dramatic difference between the

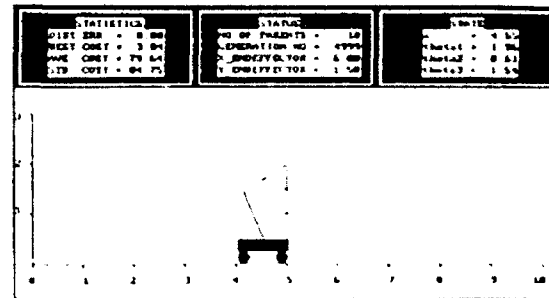


Figure 1: Converged Solution when Weighting Tip Position

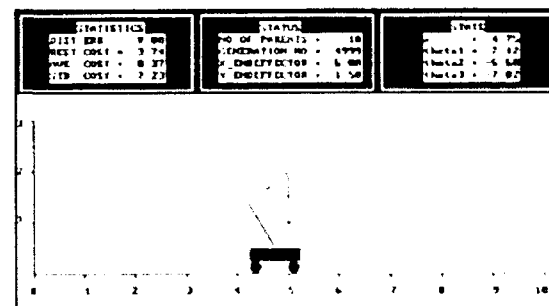


Figure 2: Converged Solution when Constraining Tip Position

two approaches can be seen in Figures 3 and 4, which show the mean and best costs for the two approaches, respectively. The difference in the horizontal scales of the two figures should be noted. The first approach took 4875 iterations to converge to the point where the value of the cost function of the best configuration was within 0.01 of its final value. The second approach, however, took only 15 iterations to reach that point. Clearly, searching for the optima using an equality constraint instead of putting a large penalty on the end-point position is by far the better approach for this problem.

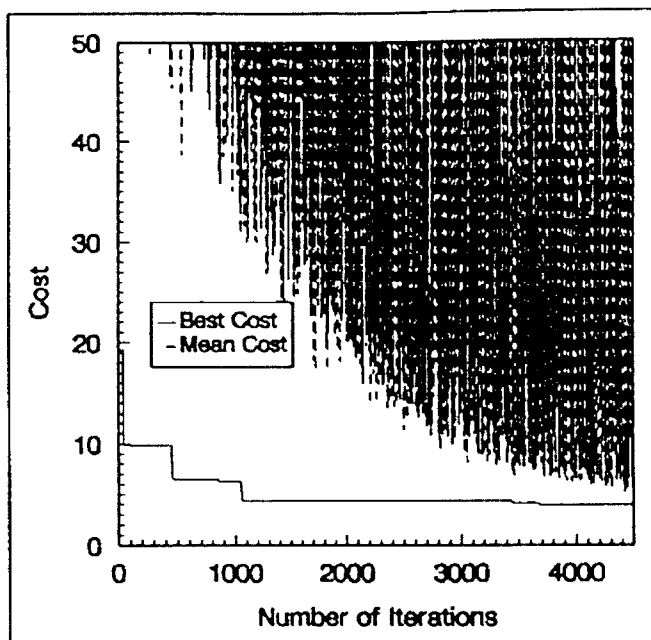


Figure 3: Best and Mean Cost when Weighting Tip Position

The faster convergence using the second approach was not the result of starting with a better set of parent vectors (vectors which resulted in the manipulator end-point being at the goal position). The same test was repeated for the unconstrained search, except that the initial population was the same as generated for the constrained search. The final configuration is shown in Figure 5, and the best and mean cost for this search are shown in Figure 6. Although the search converged to approximately the same final configuration as the other two searches, the cost is slightly higher. It took over 2200 iterations before this search found a better configuration vector than the best one from its initial population.

There are two possible causes for this dramatic improvement in convergence speed when the end-point position is constrained. First, the dimensionality of the search space has been reduced by constraining the end-point position. Second, the use of a large weight on one ele-

ment of the cost function while the other elements have much smaller weights results in a search space that is difficult to traverse. The effects of these two factors are examined in the next two sections, using a simplified model.

Large Weights in the Cost Function

Placing a large weight on one element of the cost function that represents a particularly important constraint on the system can result in a search space that is difficult to traverse. In this section, a simplified model is presented to study the effect this has on convergence.

The cost function chosen to study this effect was

$$Cost = A(x+y-10)^2 + (x-y)^2$$

The two terms of this cost function are minimized along different lines in the space of x and y . The minima of this function occurs where $x = 5$ and $y = 5$. There are no local minima in the search space.

The search space for this cost function is shown in Figure 7 for the case when $A=100$. With the large weight on the first term in the cost function, the search surface is a narrow, steep valley, where the floor of the valley is nearly flat compared to the steepness of the walls. Once an evolutionary search has found its way to the floor of the valley, it would have to mutate the solution in precisely the correct direction to find a better solution than the one it has already found.

To illustrate this, let a parent vector be $x = 6$ and $y = 4$, and let the mutation of

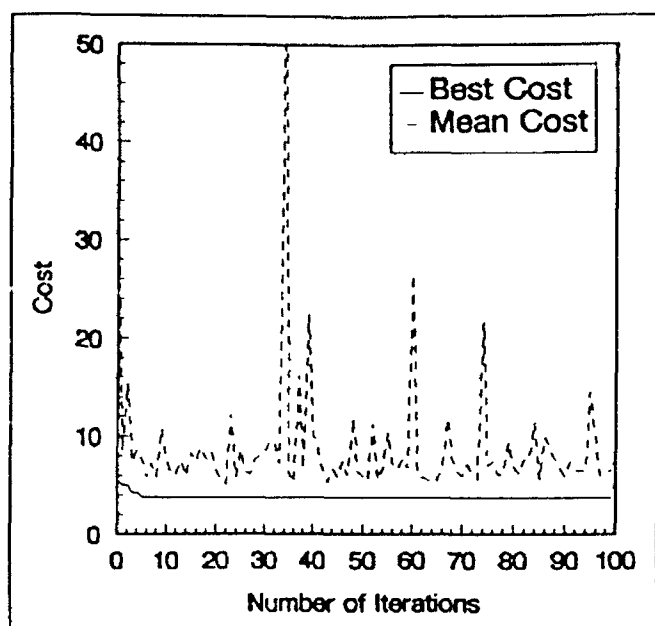


Figure 4: Best and Mean Cost when Constraining Tip Position

the y component be +0.1. If the weight A on the first term of the cost function is 100, the mutated x component must fall between 5.97 and 5.79 to result in a better solution, a range of mutations of 0.18. If, however, the weight A is 10000, the mutated x component must be between 5.91 and 5.89 to result in a better solution, a range of only 0.02. This example indicates that as the ratio of the cost function weights increases, the probability of a mutation being better than its parent decreases once the best parent vector has reached the valley floor.

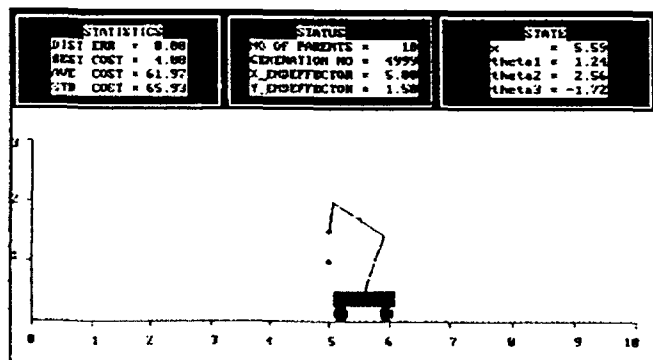


Figure 5: Best Configuration when Weighting Tip Position and a Starting Population with Tip at Goal Position

However, if the solution is assumed to satisfy the equality constraint,

$$10 = x + y$$

the search surface is reduced to the curve shown in Figure 8. This demonstrates the simplification to the search surface that results from using the equality constraint.

To study the effect of the ratio of the weights, the value of A was varied. For each value of A, ten evolutionary searches were conducted, each starting with a different parent population. Each search was continued until the best solution was within a radius of 0.01 units of the actual minima of the function. Figure 9 shows the average of the number of iterations to achieve this level of convergence as a function of the value of A. This indicates that the number of iterations required for convergence increases as the ratio of the weights (the steepness of the valley walls) increases. However, the number of iterations increases by only approximately a factor of seven as the weight ratio increases by seven orders of magnitude. For this system, the effect of the weight ratio does not increase dramatically until very large weight ratios are employed.

The difficulties of searching in these narrow, steep valleys with relatively flat floors might be overcome by augmenting the evolutionary search with a local search method. This approach has been suggested by Waagen *et al* [7], using the direction set algorithm as the local method.

Number of Parameters in Search

The other factor that increases the effi-

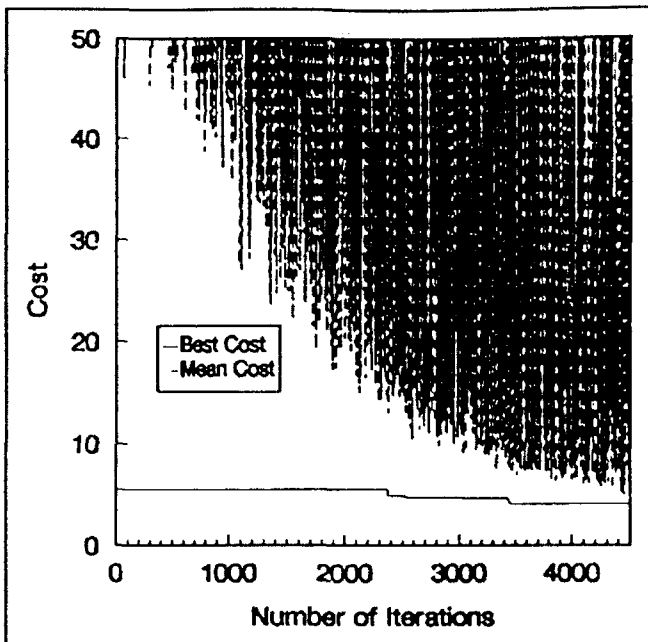


Figure 6: Best and Mean Cost when Weighting Tip Position and a Starting Population with Tip at Goal Position

ciency of the search when using an equality constraint is the reduction in the dimensionality of the search. Instead of searching for four parameters, the search is attempting to find the minima for two parameters. In this section, the effect of reducing the number of parameters is examined.

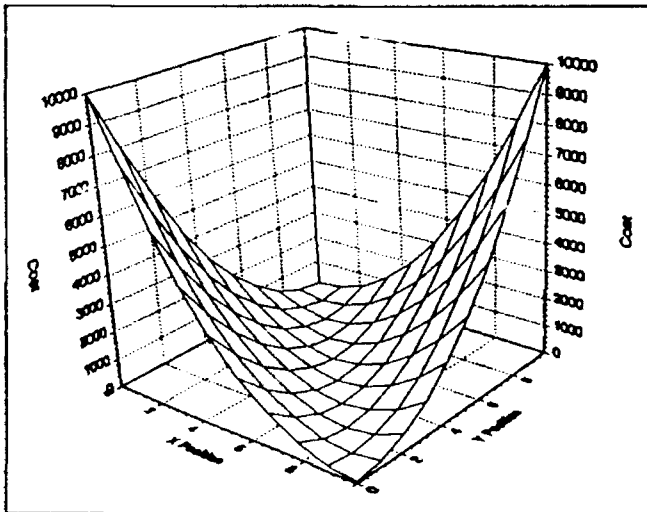


Figure 7: Search Space of Simplified Model

To study the effect of the number of parameters, the simplified model was modified. The cost function was modified to be a function of twelve variables, x_1 through x_6 and y_1 through y_6 . The cost function was specified by the three equations

$$x = \sum_{i=1}^6 x_i$$

$$y = \sum_{i=1}^6 y_i$$

$$\text{Cost} = 1000(x+y-10)^2 + (x-y)^2 + \sum_{i=1}^6 \left[\left(x_i - \frac{5}{6} \right)^2 + \left(y_i - \frac{5}{6} \right)^2 \right]$$

The minima of this cost function occurs when all the x_i and y_i are $5/6$.

First, the minima was found searching for all twelve variables. The search was repeated ten times, starting from different randomly-selected parent vectors, and the average number of iterations to converge to a solution within 0.01 units of the actual solution was determined. This process was repeated five additional times, each time constraining an additional pair of x_i and y_i to the values where the function was minimized. Figure 10 shows the average number of iterations for convergence as a function of the number of parameters which the search attempted to find. This number of iterations does not increase monotonically with the number of parameters in the search as was expected. While the general trend is upward, it is not clear why the number of iterations drop for increasing number of parameters.

It is interesting to note that the combined effect of the reduction in dimensionality and the simplification of the

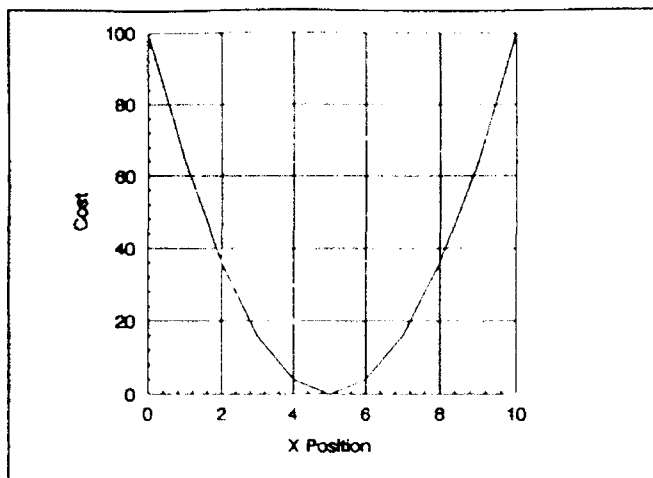


Figure 8: Search Surface with Equality Constraint

search surface for this simplified model can only account for a reduction on the order of ten times, whereas the increase in speed of convergence using the equality constraint for the mobile manipulator was on the order of one hundred times. This discrepancy might be explained by the greater complexity of the search surface for the mobile manipulator.

Conclusions

Evolutionary programming has been applied to

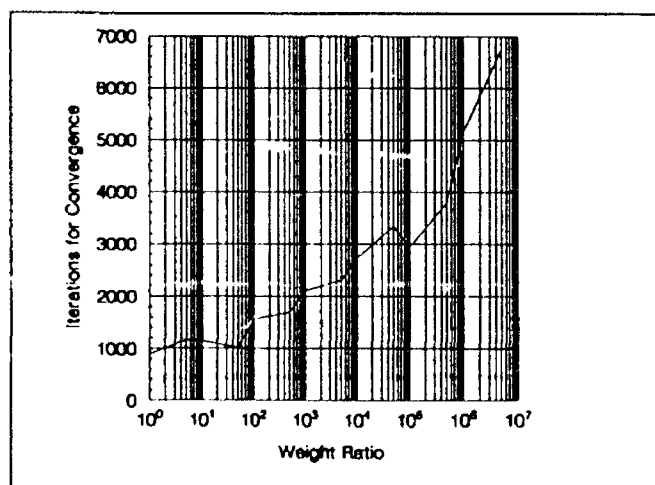


Figure 9: Average Number of Iterations for Convergence as Weight Ratio is Varied

the problem of finding the best configuration for a mobile manipulator with redundant degrees of freedom. The search was conducted both by placing a large penalty on the deviations of the end-point position from the goal position and by fixing the end-point at the goal position and only searching for two elements of the configuration vector. The search converged far faster when the end-point of the manipulator was constrained.

This increase in performance is due to two factors: the search surface is difficult to traverse when one factor of the cost function is weighted significantly more than the other; and the dimensionality of the search has been reduced. The results of an analysis of a simplified model indicate that these factors increase the convergence speed by approximately one order of magnitude, but the combined effect does not seem to actually account for the total effect seen in the simulation of the actual manipulator.

In a practical implementation of configuration optimization using EP, using the end-point constraint yields superior performance. For a general system, the dimensionality of the search can only be reduced if there is some a priori knowledge of the minima of the system. If possible, one should incorporate the constraints to speed convergence.

References

1. *Teleoperated Robotics in Hostile Environments*, H.L. Martin and D.P. Kubans, Eds., Robotics International of SME, Dearborn, MI., 1985
2. F.G. Pin, and J-C. Culioli, "Multi-Criteria Position and Configuration

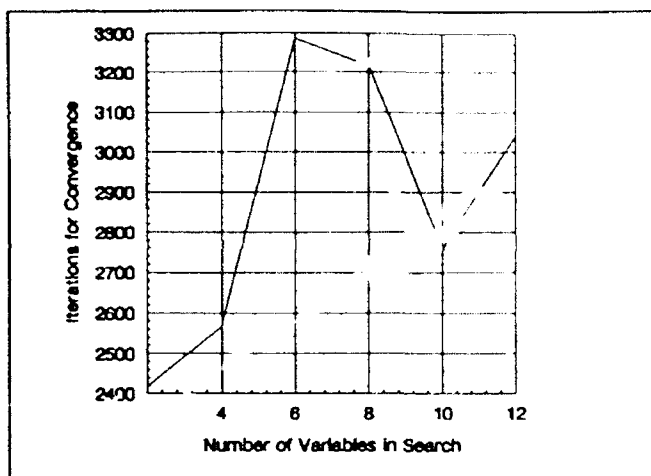


Figure 10: The Number of Iterations for Convergence of the Search for Varying Numbers of Parameters in the Search

Optimization for Redundant Platform / Manipulator Systems", IEEE Int. Workshop on Intelligent Robots and Systems, IROS '90.

3. J. Burdick, and H. Seraji, "Characterization and Control of Self Motions in Redundant Manipulators", JPL Invention Report, NPO-17965/7463, 1991.

4. W.F. Carriker, P.K. Khosla, and B.H. Krough, "The Use of Simulated Annealing to Solve the Mobile Manipulator Path Planning Problem", 1990 IEEE Conference on Robotics and Automation.

5. J. McDonnell, W. Page, and B. Andersen, "Configuration Optimization for Platform-Manipulator Systems Using Evolutionary Programming", 1991 ASME Conference on Neural Networks, San Diego, CA.

6. D. Fogel, *System Identification Through Simulated Evolution: A Machine Learning Approach*, Ginn Press, Needham, MA, 1991.

7. D. Waagen, P. Diercks, and J. McDonnell, "The Stochastic Direction Set Algorithm: A hybrid technique for finding function extrema",

First Annual Conference on Evolutionary Programming, San Diego, CA, 1992.

DTIC QUALITY INSPECTED 2

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	20